FortiGate on Microsoft Azure

Gabriel Kälin, Fortinet

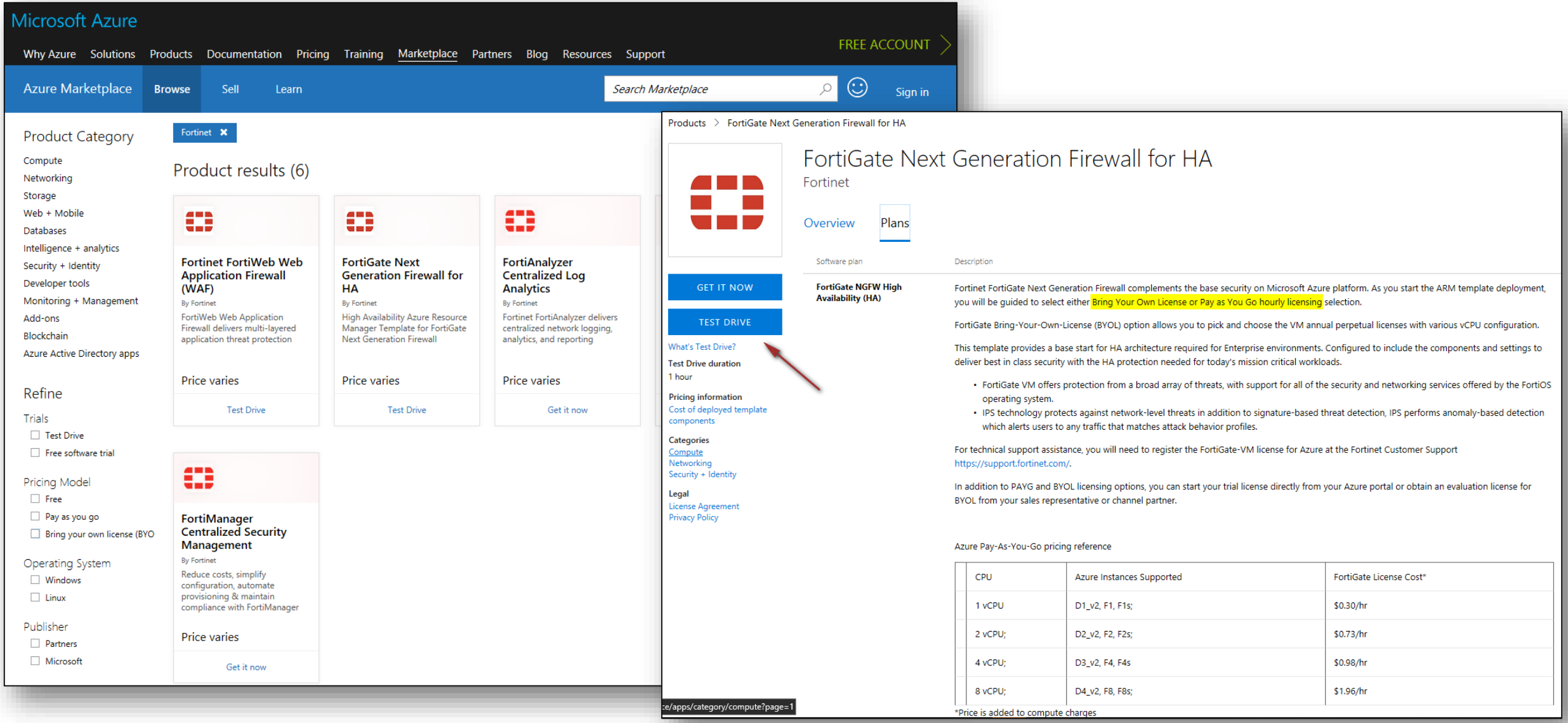
October 24, 2018

Agenda

- Scenario 1: Single FortiGate on Azure for IPsec VPN to on-premise
  - » Using the FortiGate to Azure Connector
  - » Automation with template deployment
- Scenario 2: HA Active-Passive
- Scenario 3: Network Segmentation
- Scenario 4: Cloud DMZ with HA Active-Active

Scenario 1: Single FortiGate on Azure for IPsec VPN to on-premise

Azure Marketplace



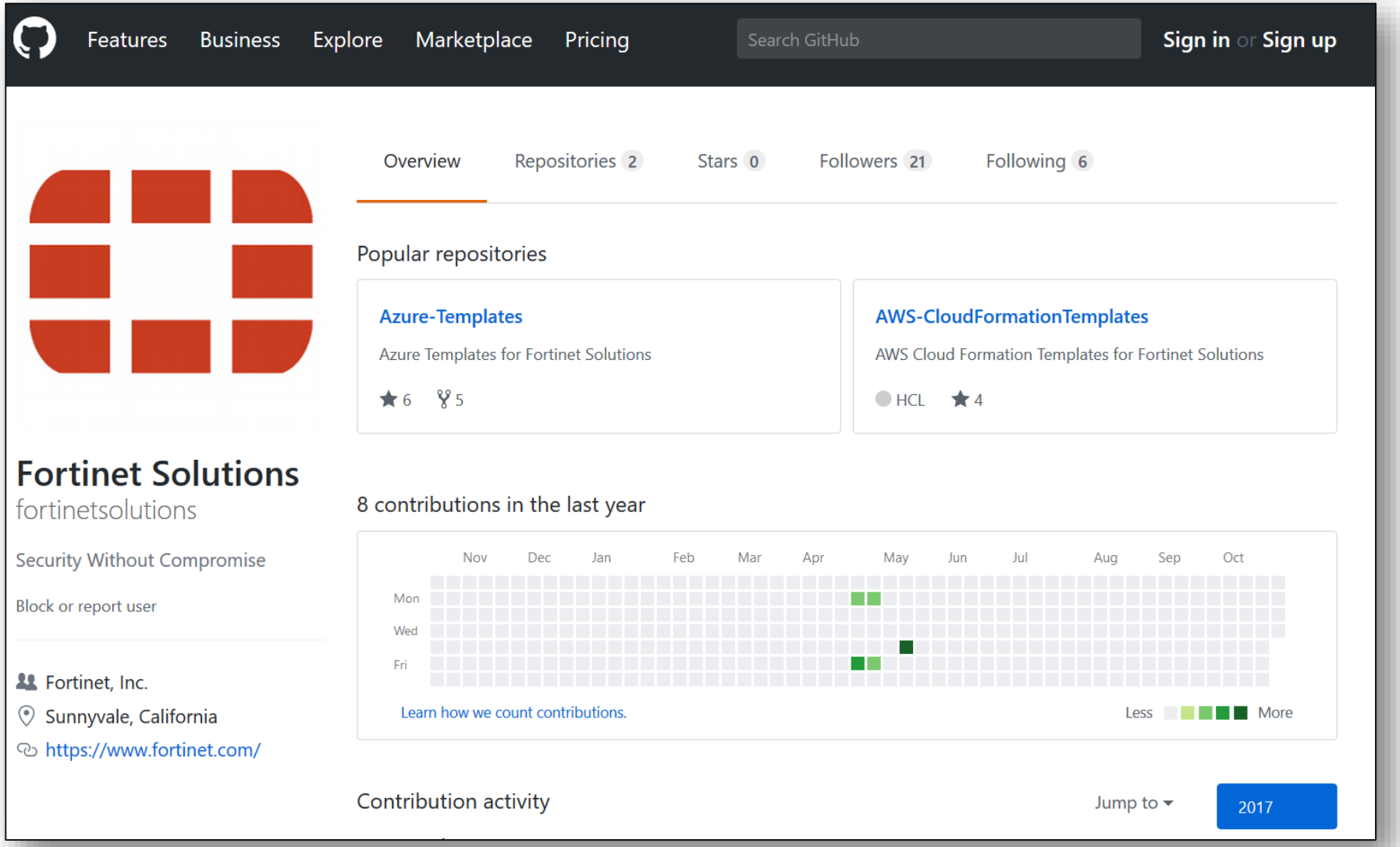
How to Choose the Right FortiGate Model

- Use v-Series FortiGate VM: FG-VMxxV
  - » VDOM use restricted by number of NICs.
  - » VDOM upgrade possible
  - » Management VDOM will be available by default
- New with FortiOS 6.0.2 and later: FortiGate VM can run on larger instances w.r.t. vCPU than the license would allow. Excess vCPUs will be ignored.

Instance type	vCPU	Max NIC	FortiGate minimum order (BYOL)
F1	1	2	FG-VM01 or FG-VM01v
F2, F2s, F2s_v2	2	2	FG-VM02 or FG-VM02v
F4, F4s, F4s_v2	4	4 (2 for F4s_v2)	FG-VM04 or FG-VM04v
F8, F8s, F8s_v2	8	8 (4 for F8s_v2)	FG-VM08 or FG-VM08v
F16, F16s, F16s_v2	16	8 (4 for F16s_v2)	FG-VM16 or FG-VM16v

https://docs2.fortinet.com/vm/azure/fortigate/6.0/about-fortigate-for-azure/6.0.0/271726/instance-type-support

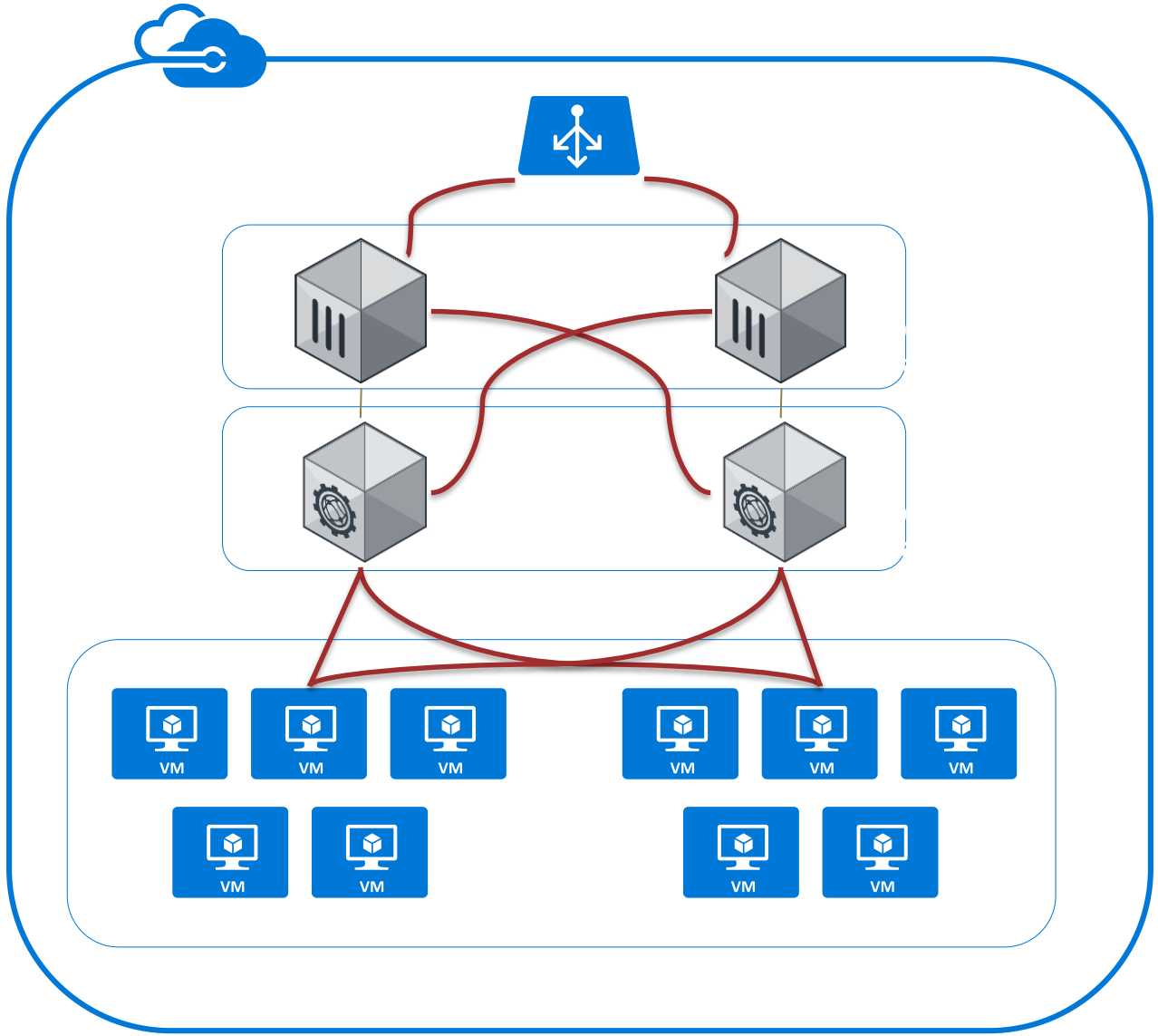
Custom Sample Templates from Fortinet



https://github.com/fortinetsolutions

FortiGate on Azure – Living with playground rules

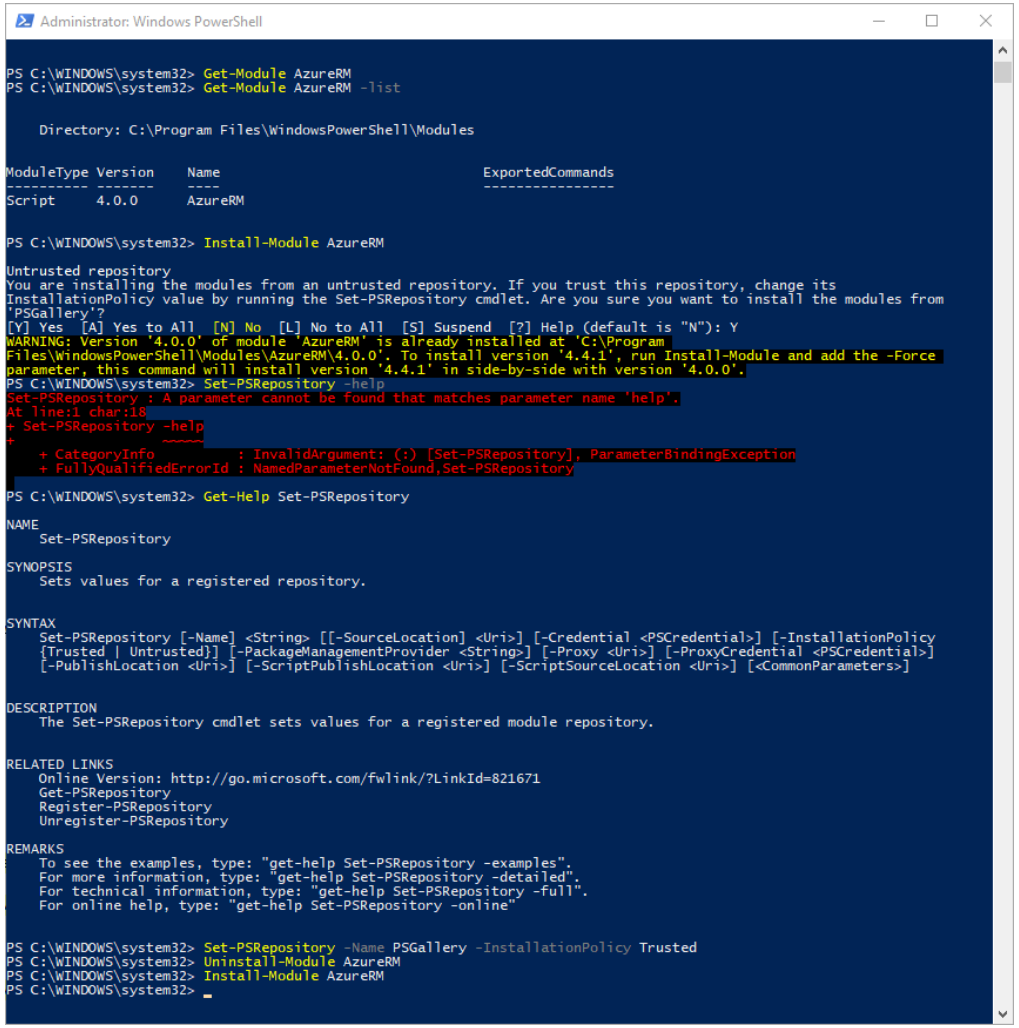
Microsoft Azure



1. You may not manipulate layer 2
2. You may not use Multicast
3. You may not reassign IP addresses
4. You may create one and only one next hop per custom route.

Azure with PowerShell – Getting Started

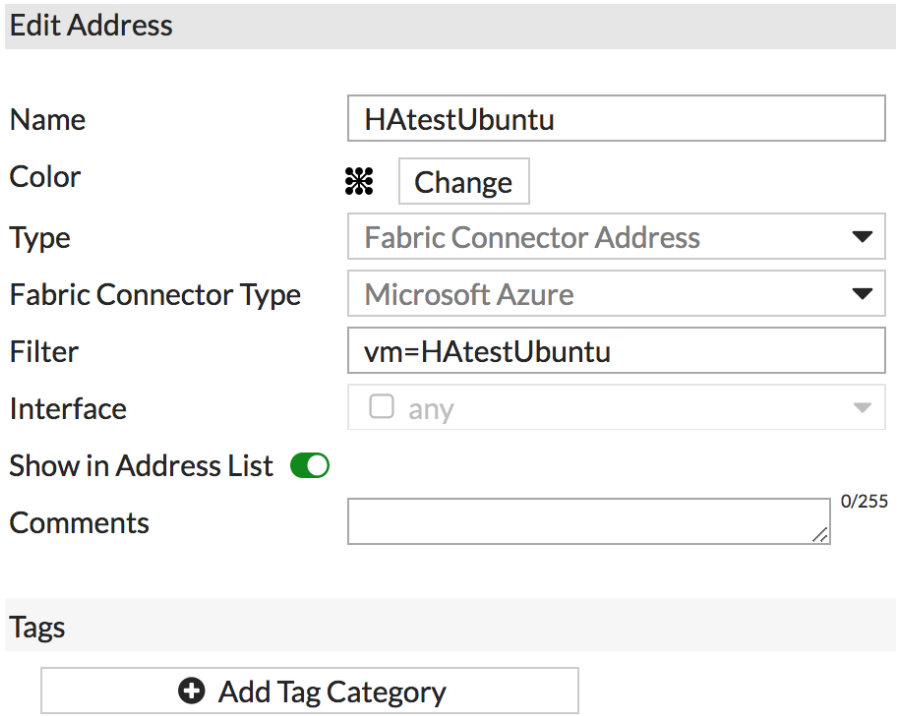
- If you're starting with PowerShell (PS):
  - » Trust the repository "PSGallery" to avoid warnings
    - `Set-PSRepository -Name PSGallery -InstallationPolicy Trusted`
  - » Allow execution of local scripts, e.g.
    - `Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy remotesigned`
- Install Azure Resource Manager module
  - » `Get-Module AzureRM`
  - » `Import-Module AzureRM`
- Check installed version of AzureRM
  - » `Get-Module AzureRM –ListAvailable`
- If there is a new version available
  - » `Update-Module AzureRM`

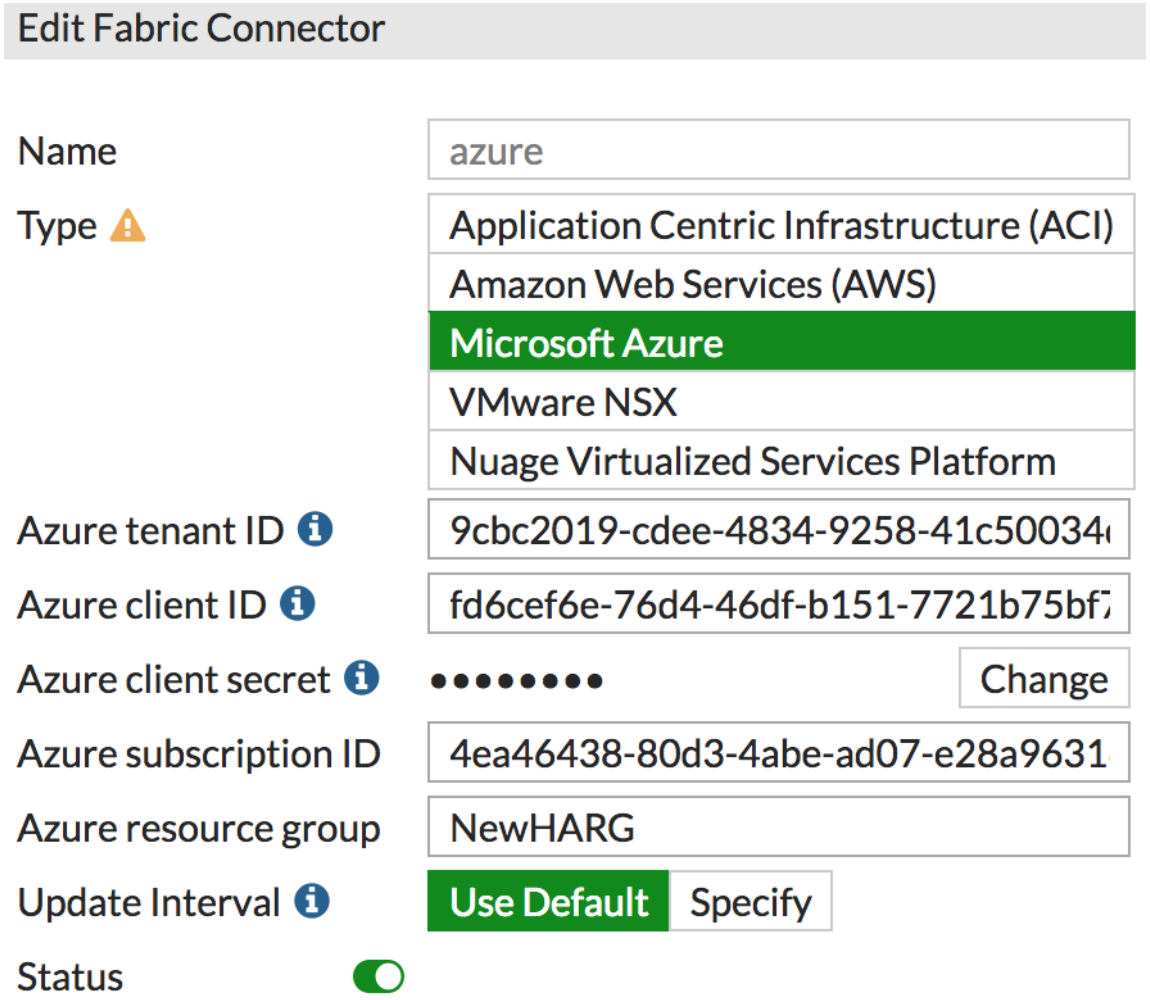


Using Azure Connectors

Security Fabric connector for Azure

- Part of FortiOS 6.0 multi-cloud Security Fabric support
- Also to update dynamic addresses





Azure SDN Connector supported filters

- set filter 'vm=<name>'
- set filter 'securitygroup=<nsg name>'
- set filter 'vnet=<virtual network name>'
- set filter 'subnet=<subnet name>'
- set filter 'vmss=<vmss name>'
- set filter 'tag.<key>=<value>’
- set filter ‘vm=<name> & vnet=<virtual network name>’
- set filter ‘vm=<name> | vm=<name>’

- Current implementation will limit the use to one Azure subscription, and only resources in use and associated with running VMs
- FortiOS 6.2 will enable multiple connectors of the same type

License and Config Bootstrapping

Deployment with custom data in Template with PowerShell

How to Format Custom Data

- Requires base64 encoded multipart/mixed content
- Use/insert customData attribute in osProfile section of template.json

Example

```
Content-Type: multipart/mixed; boundary="==========0086047718136476635=="
MIME-Version: 1.0

--========0086047718136476635==
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="config"

config sys global
    set admintimeout 120
    set hostname "mycloudinit"
    set timezone 26
end

--========0086047718136476635==
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="FGVM010000156211.lic"

-----BEGIN FGT VM LICENSE-----
QAAAAOckKptgXQ0Mvc/3UNfiIqGohp8MGWOCWh+wSqzcfQV/7e5suEXkls5Uvm9b
ww/aX3dxCvpOFVRTDmadfdfbHwFgAAAAC+a0IlJRT+b72wqY+jUzH7/AW4IlAhIU
f4EPIKgrKoFxdIazYhBXTYwOUeokRx0ySNqMlYzfdWEa26Gq9/ecdt1FKYKeJ2lf
ptbePSNd2w8emQbk5yvlm4i3npJT+YgO
-----END FGT VM LICENSE-----

--========0086047718136476635==--
```

Insert customData Field Into Template

C:\Users\gkaelin\Documents\Channel\Distributoren\ALSO\2018-10-24 FortiOS 6.0 Workshop - Azure\automation\preparation\template-fgt-cloudinit.json

```
20                  "description": "Password for the FortiGate virtual appliance."
21              }
22          },
23          "customData": {
24              "type":"string",
25              "metadata": {
26                  "description": "check Mantis 0449239 for data format"
27              }
28          },
29          "FortiGateName": {
30              "type": "string",
31              "metadata": {
32                  "description": "Name for FortiGate virtual appliance."
33              }
462                     "id": "[variables('compute_AvailabilitySet_FG_Id')]"
463                 },
464                 "osProfile": {
465                     "computerName": "[parameters('FortiGateName')]",
466                     "adminUsername": "[parameters('adminUsername')]",
467                     "adminPassword": "[parameters('adminPassword')]",
468                     "customData": "[parameters('customData')]"
469                 },
470                 "storageProfile": {
```

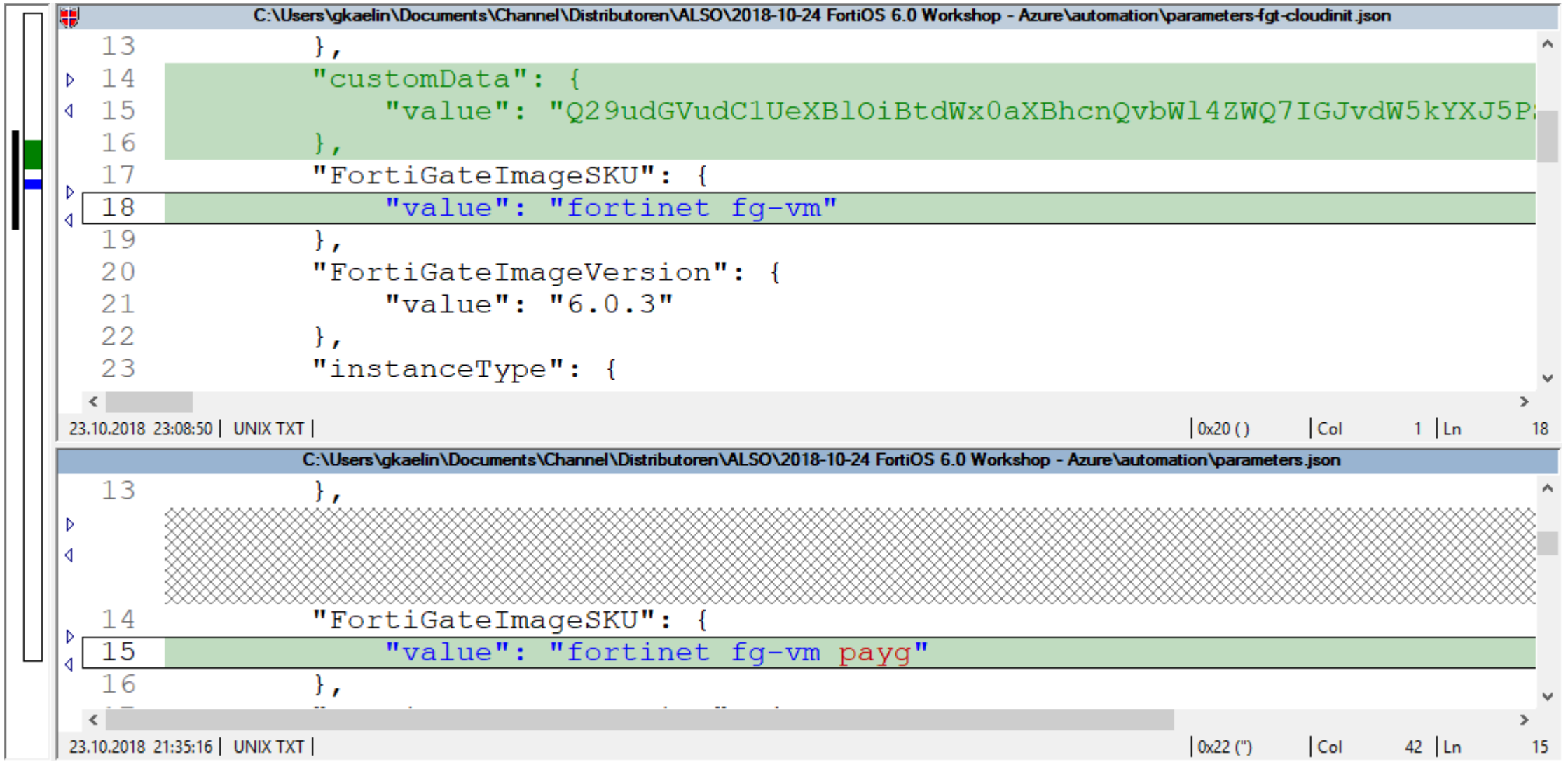
23.10.2018 23:11:46 | UNIX TXT | | 0x20 ( ) | Col 1 | Ln 23

C:\Users\gkaelin\Documents\Channel\Distributoren\ALSO\2018-10-24 FortiOS 6.0 Workshop - Azure\automation\preparation\template.json

```
25              "metadata": {
26                  "description": "Name for FortiGate virtual appliance."
27              }
456                     "id": "[variables('compute_AvailabilitySet_FG_Id')]"
457                 },
458                 "osProfile": {
459                     "computerName": "[parameters('FortiGateName')]",
460                     "adminUsername": "[parameters('adminUsername')]",
461                     "adminPassword": "[parameters('adminPassword')]"
462                 },
463                 "storageProfile": {
464                     "imageReference": {
```

23.10.2018 12:35:34 | UNIX TXT | | - | Col 1 | Ln

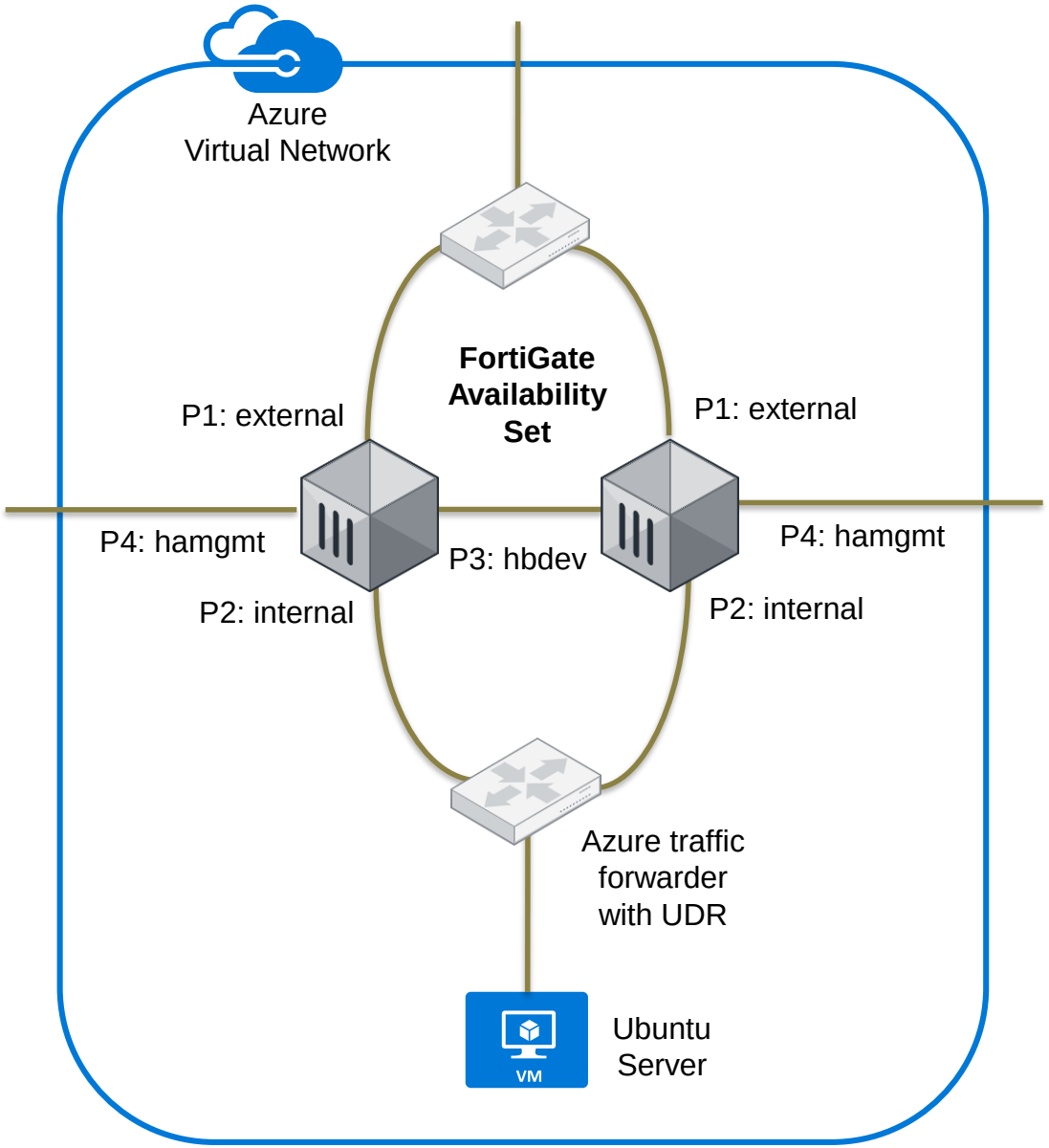
Define customData Value in Parameters File



Scenario 2: HA Active-Passive

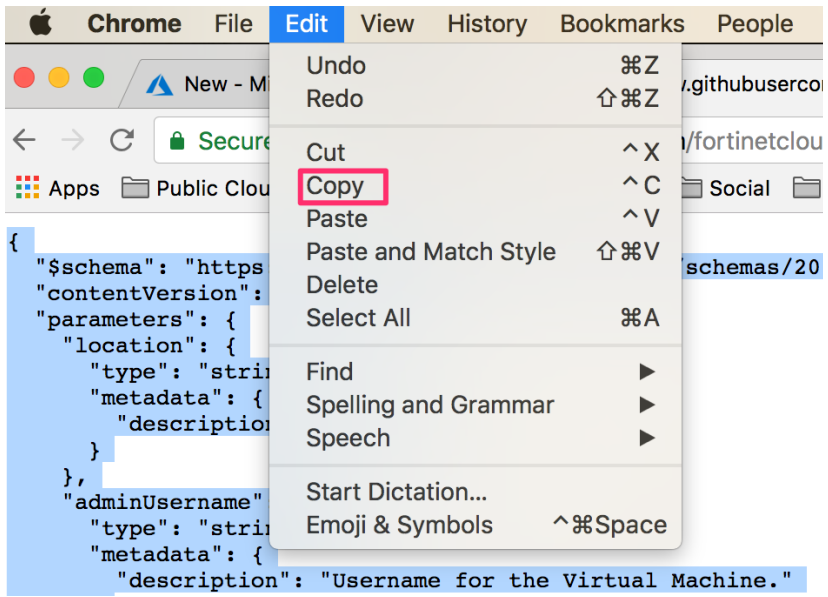
New Active/Standby HA

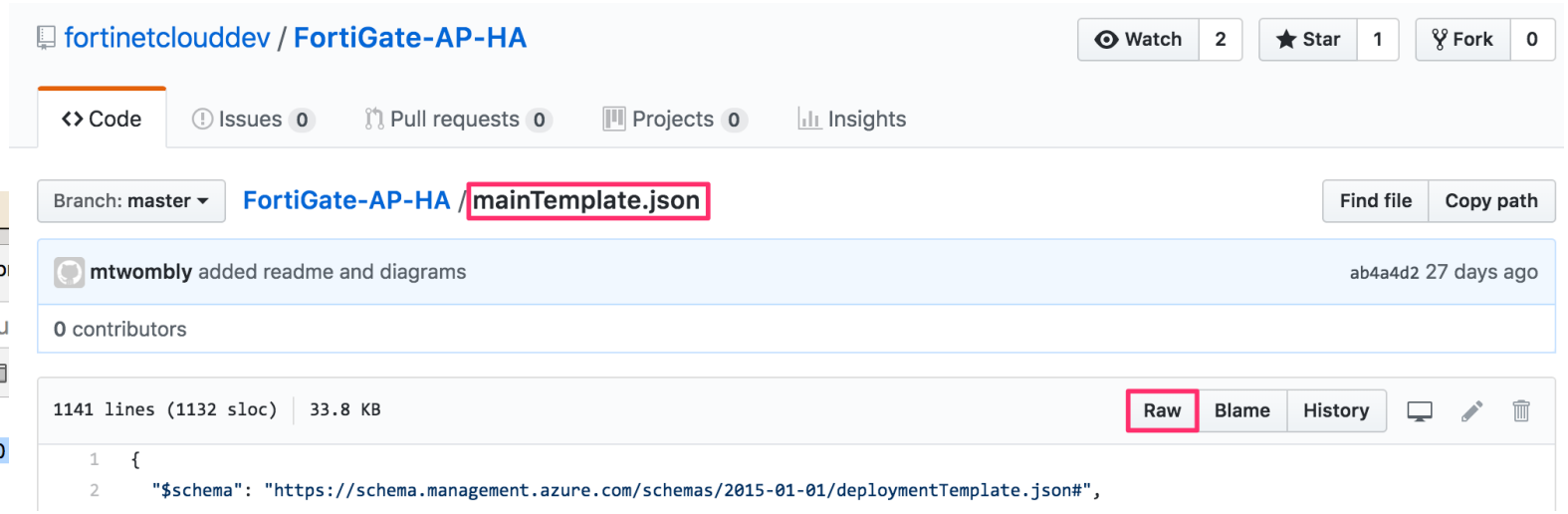
- FortiOS 6.0.0 and 5.6.4
- Minimum three interfaces, better four
  - » Heartbeat and management interfaces in (hidden) sys VDOM, unusable for productive traffic
  - » Management interface for accessing firewalls
- Inside one Azure Region
- Additionally cloud-init support for Azure is now native to the cloud
- FortiGate VM for Azure also supports bootstrapping
  - » Will enable auto-scaling at some point



Fetch latest FortiGate Template from GitHub

- https://github.com/fortinetclouddev/FortiGate-AP-HA
  - » Click ‘MainTemplate.json’
  - » Click ‘Raw’
  - » Select all and ‘Copy’





New HA CLI

- **HA Sync** must use unicast IP to sync, cannot use L2
- **Config sync** also now working over unicast IP addresses
- **Failover Mechanism** by sending commands directly to Azure
  - » Move public IP addresses
  - » Manipulate UDRs
- Failover times **unpredictable**
  - » Depends of number of items to re-write
  - » Serial change, not parallel

```
config system ha
    set group-name "Test"
    set mode a-p
    set hbdev "port3" 100
    set session-pickup enable
    set session-pickup-connectionless enable
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "port4"
            set gateway 10.1.3.1
        next
    end
    set override disable
    set priority 255
    set unicast-hb enable
    set unicast-hb-peerip 10.1.2.5
end
```

Tips for Successful Failover Testing (1/2)

- Static IP addresses make sense
  - » Keep config in sync in Azure and FortiOS
- ‘config system sdn-connector’ files are not synced and require independent settings for NIC and UDR next-hop
- Inbound traffic requires two Virtual IP elements
  - » Firewall IP address config is not synced
- Remember static routes to internal networks
- Set override disable

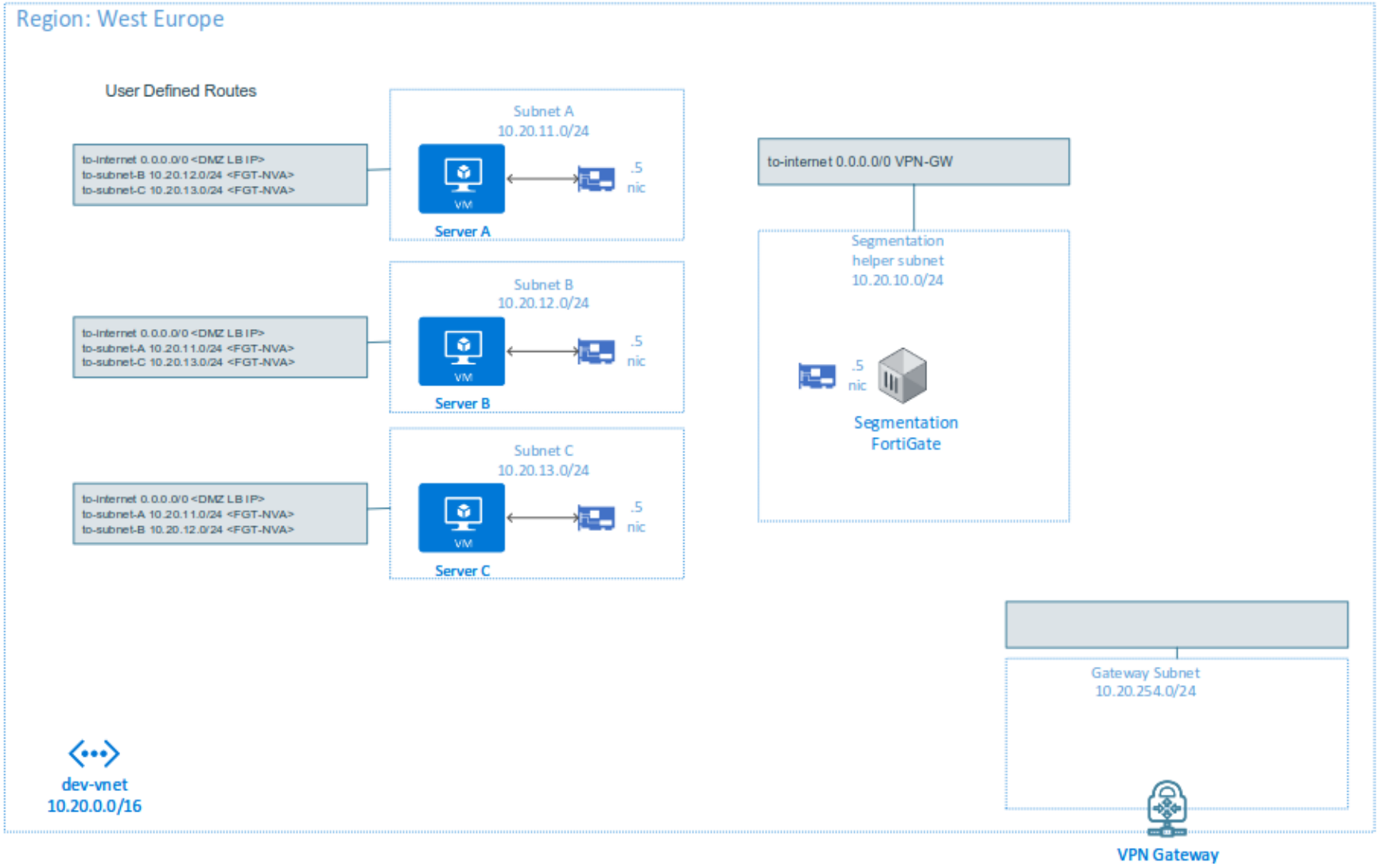
Tips for Successful Failover Testing (2/2)

- diagnose sys ha checksum show
- diagnose debug application azd -99
- diagnose test application azd -1
  1. show HA stats
  2. SDN api test
  3. HA api test
  4. filter list test
  99. restart
- ‘HA api test‘ is NOT a FGCP failover, leading to the twilight zone ☺
  - » Test with a real failure
- Make sure you have independent IP addresses. Remember, they are not synced ;)
- Headache with synced config, including node-specific IP addresses?

```
config system ha
        set sync-config disable
end
```

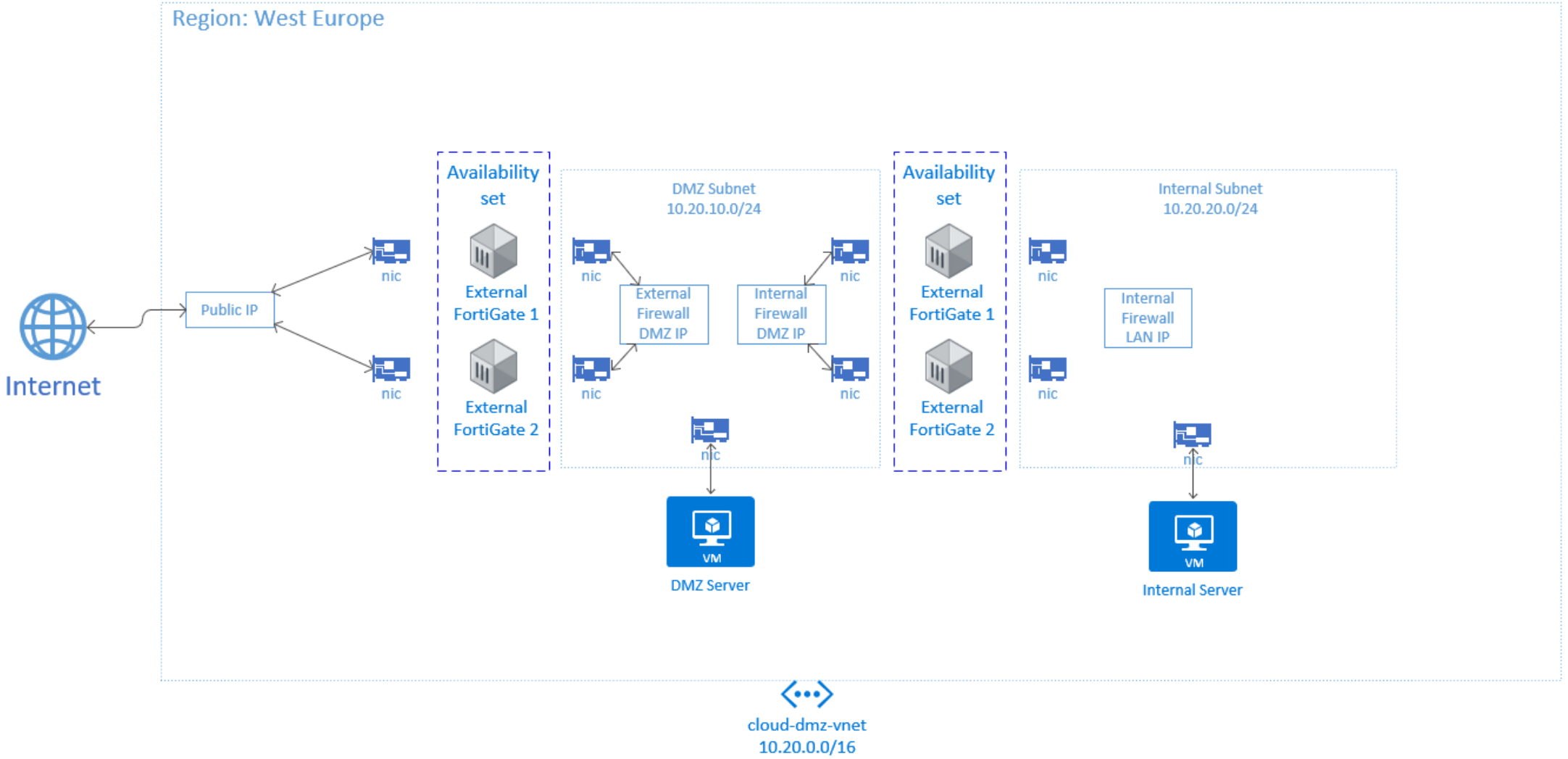
Scenario 3: Network Segmentation

Network Segmentation – Single Instance FortiGate

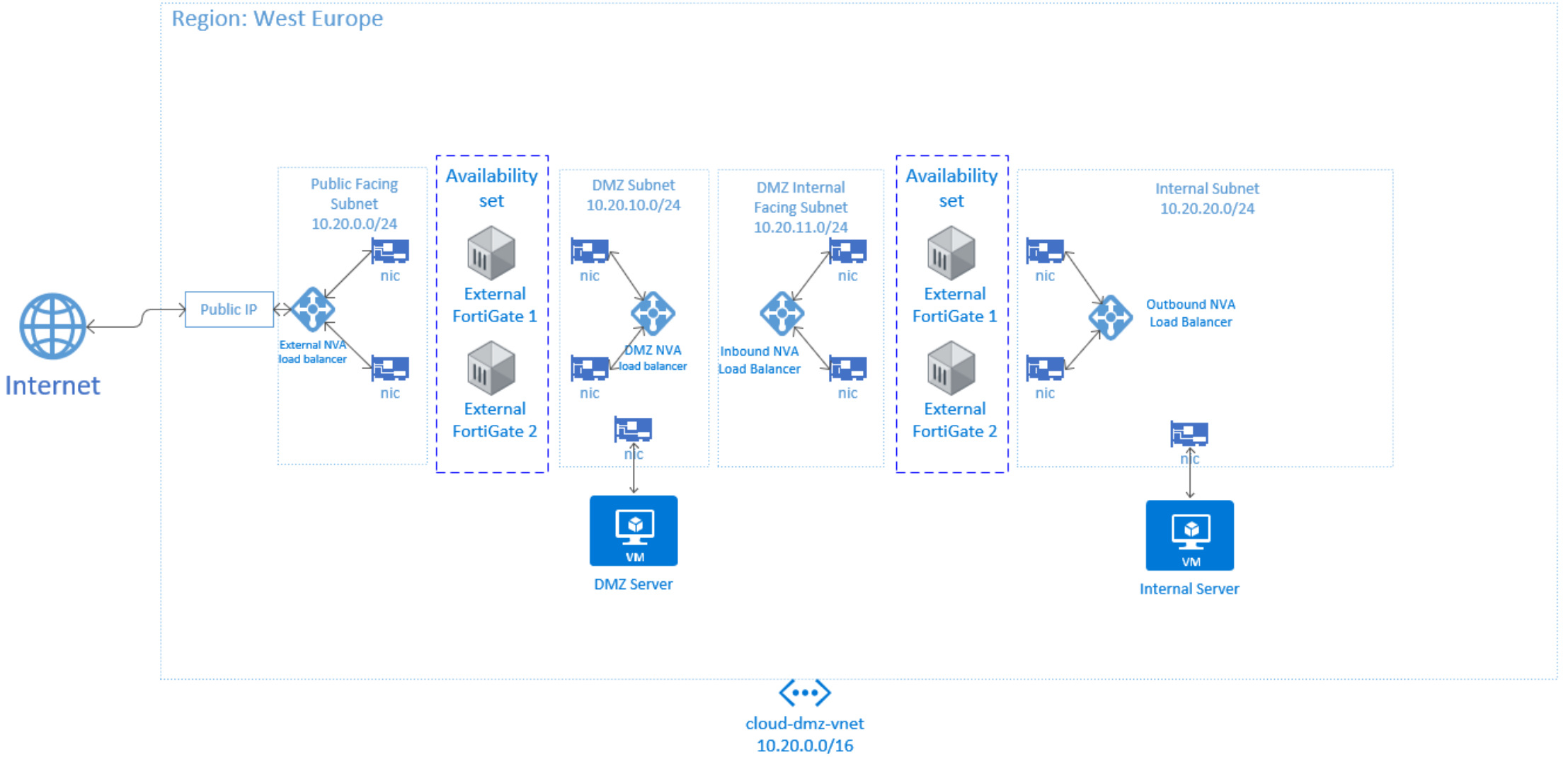


Scenario 4: Cloud DMZ with HA Active-Active

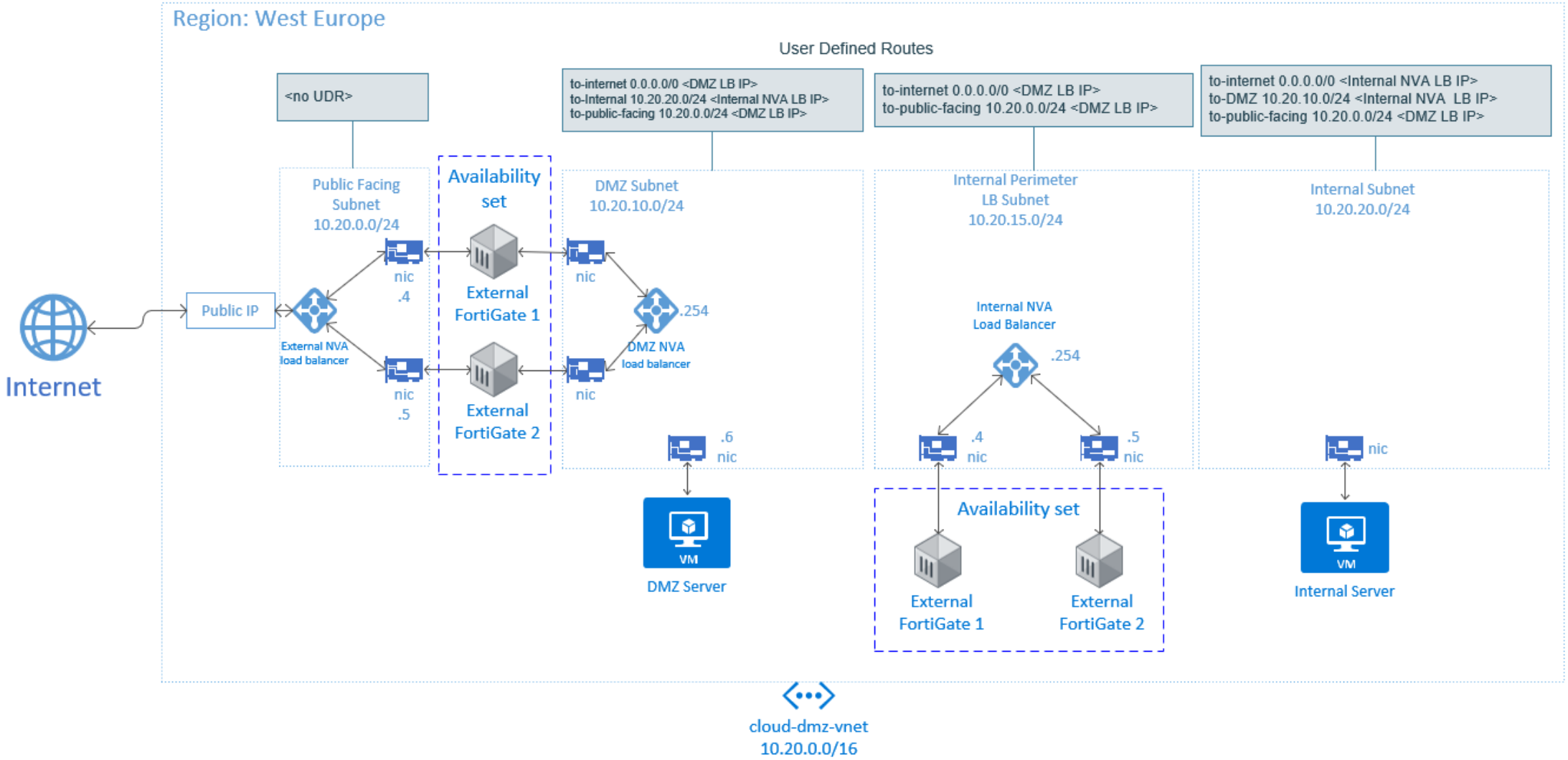
Wishful DMZ


Region: West Europe
Internet
Public IP
nic
nic
Availability set
External FortiGate 1
External FortiGate 2
DMZ Subnet
10.20.10.0/24
nic
External Firewall DMZ IP
Internal Firewall DMZ IP
nic
nic
nic
nic
VM
DMZ Server
Availability set
External FortiGate 1
External FortiGate 2
Internal Subnet
10.20.20.0/24
nic
nic
Internal Firewall LAN IP
nic
VM
Internal Server
cloud-dmz-vnet
10.20.0.0/16

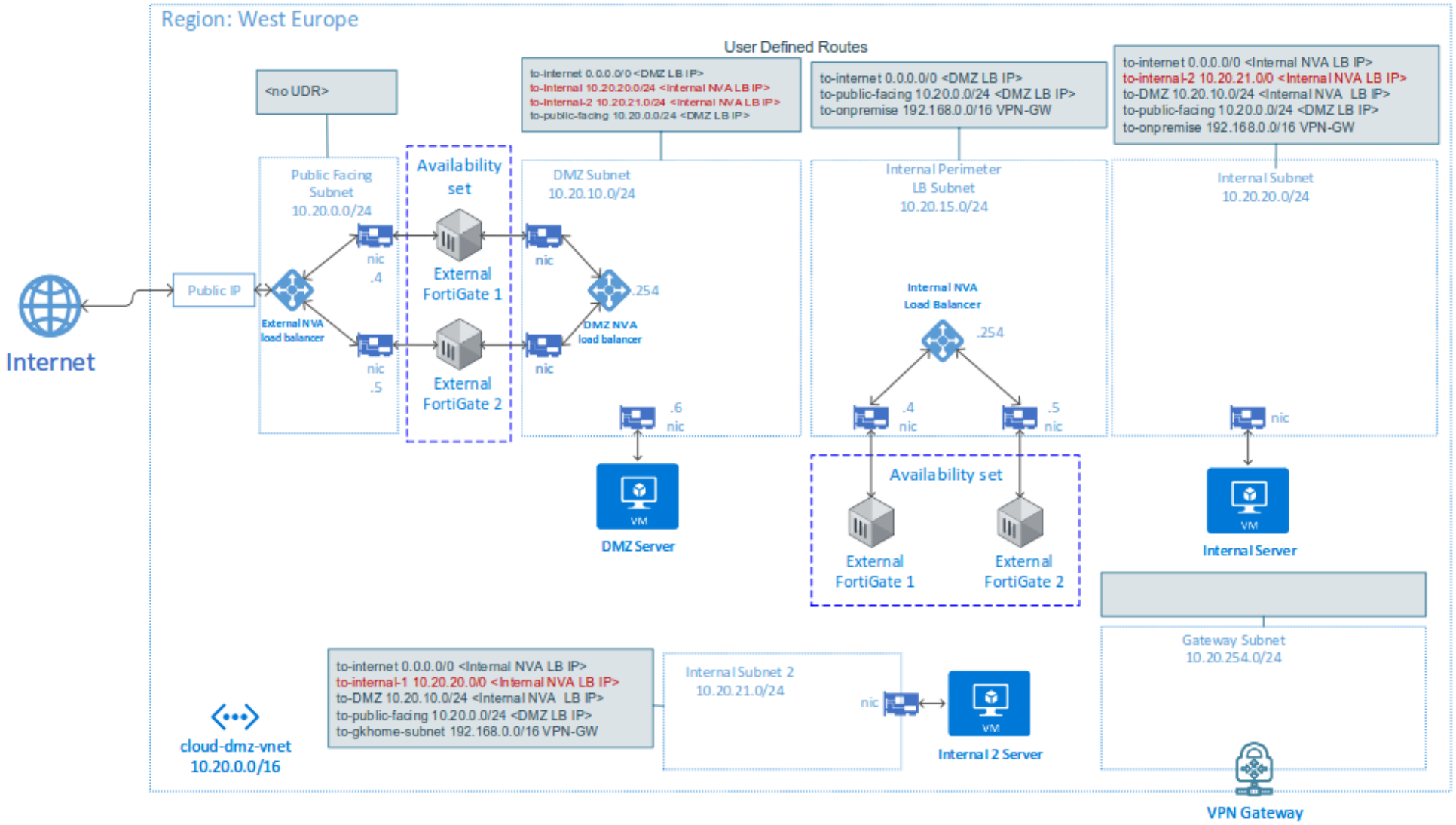
One Step Towards Reality – Load Balancers Everywhere



Validated Design – Single Arm Internal FortiGate

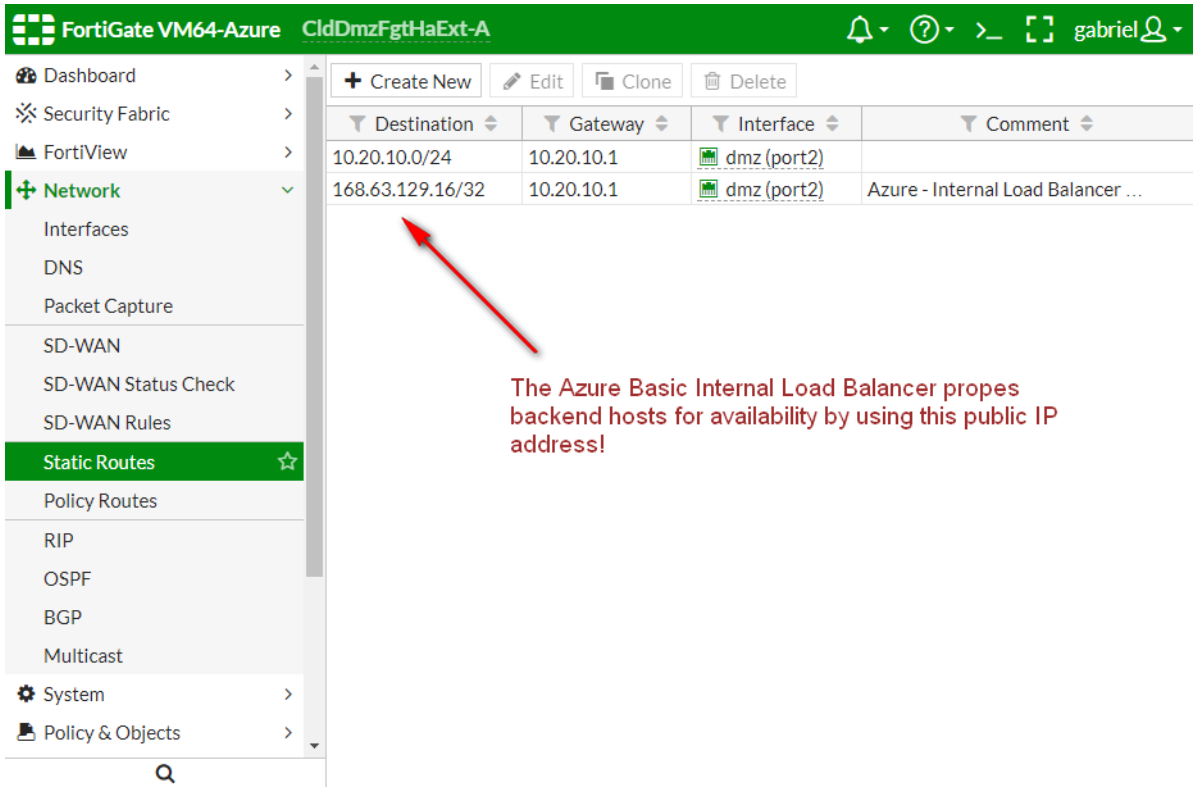


Cloud DMZ with Internal Segmentation



External FortiGate Configuration – Routing

- There seems to be no way to check the backend probe status for Internal Load Balancers (Basic).
- Check probe traffic on FortiGate:




```
CldDmzFgtHaExt-A # diagnose sniffer packet port2 'port 22'
interfaces=[port2]
filters=[port 22]
2.441215 168.63.129.16.62861 -> 10.20.10.4.22: syn 1157932754
2.443124 168.63.129.16.62862 -> 10.20.10.4.22: syn 914950808
4.442631 168.63.129.16.62923 -> 10.20.10.4.22: syn 521302262
4.442654 168.63.129.16.62922 -> 10.20.10.4.22: syn 739689286
7.442191 168.63.129.16.62922 -> 10.20.10.4.22: syn 739689286
7.443232 168.63.129.16.62923 -> 10.20.10.4.22: syn 521302262
```

CldDmzFgtHaExt-A # diagnose sniffer packet port2 'port 22'

FortiGate Configuration – Session Sync with FGSP

External FortiGate A

```
CldDmzFgtHaExt-A # show system cluster-sync
config system cluster-sync
    edit 1
        set peerip 10.20.10.5
        set syncvd "root"
    next
end

CldDmzFgtHaExt-A # show system ha
config system ha
    set hbdev "port2" 50
    set session-pickup enable
    set session-pickup-nat enable
    set override disable
end

CldDmzFgtHaExt-A #
```

External FortiGate B

```
CldDmzFgtHaExt-B # show system cluster-sync
config system cluster-sync
    edit 1
        set peerip 10.20.10.4
        set syncvd "root"
    next
end

CldDmzFgtHaExt-B # show system ha
config system ha
    set hbdev "port2" 50
    set session-pickup enable
    set session-pickup-nat enable
    set override disable
end

CldDmzFgtHaExt-B #
```

Testing Session Sync

- Diagnostic Commands:
  - » # diag debug application hasync -1
  - » # diag debug application hatalk -1
  - » # diag debug application hamonitord -1
  - » unix $ openssl s_client -connect
    [e.g.] 194.40.217.50:443
  - » fgt # diag sys session list
  - » fgt # diag sys session sync

- Example (in an AWS environment with different IP addresses than the rest in these slides ☺)

```
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID: 2ACAB49D279011C575E28025C0FAECC9F5BA0C4FDB72C47DCD0EDEF5175E931E
    Session-ID-ctx:
    Master-Key: B6C1279D39B66D5908F7FA8911D555C4C50F981D2E25029CC89281290DFA07FA3
59F140BBB26456ABD935C446366F3F0
    Key-Arg   : None
    Krb5 Principal: None
    PSK identity: None
    PSK identity hint: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - aa 04 eb 1e 5d f0 f5 a4-e9 c5 93 04 42 d3 17 1d   ....]....... B...
    0010 - ed 8b 81 19 d6 d5 36 a5-81 8b 16 b3 d0 1d 4d 10   ......6.......M.
    0020 - c5 7c 3a 01 48 cf a8 fc-2f b0 7c 75 f4 d7 d2 07   .|:.H.../.|u....
    0030 - 86 ed b4 4b 8c aa 19 05-58 08 3e 31 ab a8 28 b9   ...K....X.>1..(.
    0040 - 39 4e a2 ab 0d e2 bd b0-ba 95 e3 b1 9e 17 14 3d   9N.............=
    0050 - e8 d5 eb d0 7f 35 1e a5-56 80 26 f9 67 65 95 53   .....5..V.&.ge.S
    0060 - 21 87 c5 57 d0 51 7c 79-65 0c 99 b7 05 a7 11 98   !..W.Q|ye.......
    0070 - b0 ef c3 28 04 e6 00 a9-90 d5 ef 3d 0d 21 a1 e0   ...(.......=.!..
    0080 - 3c ea c4 ce 7f f8 6c d2-41 f4 65 b4 88 8b 02 49   <.....l.A.e....I
    0090 - 3d 58 cc fa 2c a6 e2 f3-56 34 e8 93 95 7c fc f7   =X..,...V4...|..
    00a0 - 25 f1 38 e6 b0 f5 7a 85-fe a0 76 21 3b 31 32 86   %.8...z...v!;12.

    Start Time: 1510775046
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
---
```

session still active

```
AWS-fgt1a # diag sys session list

session info: proto=6 proto_state=01 duration=4 expire=3595 timeout=3600 flags=00000000 sock
flag=00000000 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty synced
statistic(bytes/packets/allow_err): org=839/8/1 reply=5748/7/1 tuples=2
tx speed(Bps/kbps): 196/1 rx speed(Bps/kbps): 1342/10
orgin->sink: org pre->post, reply pre->post dev=4->3/3->4 gwy=172.30.0.1/172.30.10.171
hook=post dir=org act=snat 172.30.10.171:38500->194.40.217.50:443(172.29.0.1:38500)
hook=pre dir=reply act=dnat 194.40.217.50:443->172.29.0.1:38500(172.30.10.171:38500)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=000002ee tos=ff/ff app_list=0 app=0 url_cat=0
dd_type=0 dd_mode=0
total session 1

AWS-fgt1a # diag sys session sync
sync_ctx: sync_started=1, sync_tcp=1, sync_others=0,
sync_expectation=0, sync_redir=0, sync_nat=1, stdalone_sesync=1.
sync: create=11:0, update=42, delete=0:0, query=0
recv: create=0:0, update=0, delete=0:0, query=0
ses pkts: send=0, alloc_fail=0, recv=0, recv_err=0 sz_err=0
udp pkts: send=53, recv=0
nCfg_sess_sync_num=5, mtu=1500
sync_filter:
        1: vd=0, szone=0, dzone=0, saddr=0.0.0.0:0.0.0.0, daddr=0.0.0.0:0.0.0.0, sport=0-655
35, dport=0:65535

AWS-fgt1a #
```

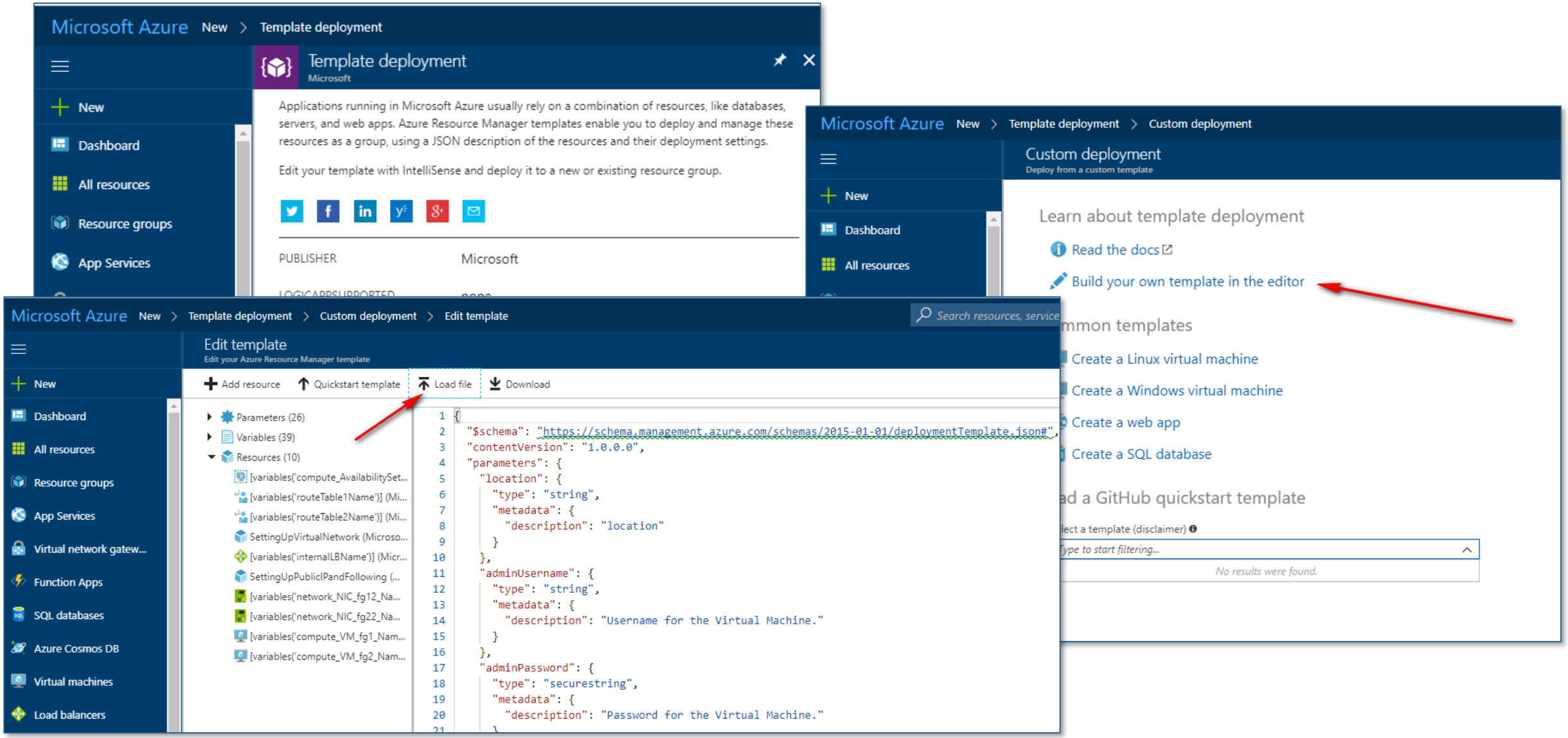
```
AWS-fgt1b # AWS-fgt1b # diagnose sys session list

session info: proto=6 proto_state=01 duration=14 expire=3585 timeout=3600 flags=00000000 sockf
lag=00000000 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=4->3/3->4 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 172.30.10.171:38500->194.40.217.50:443(172.29.0.1:38500)
hook=pre dir=reply act=dnat 194.40.217.50:443->172.29.0.1:38500(172.30.10.171:38500)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=000002ee tos=ff/ff app_list=0 app=0 url_cat=0
dd_type=0 dd_mode=0
total session 1

AWS-fgt1b # diagnose sys session sync
sync_ctx: sync_started=1, sync_tcp=1, sync_others=0,
sync_expectation=0, sync_redir=0, sync_nat=1, stdalone_sesync=1.
sync: create=0:0, update=0, delete=0:0, query=0
recv: create=9:0, update=34, delete=0:0, query=0
ses pkts: send=0, alloc_fail=0, recv=0, recv_err=0 sz_err=0
udp pkts: send=0, recv=43
nCfg_sess_sync_num=5, mtu=1500
sync_filter:
        1: vd=0, szone=0, dzone=0, saddr=0.0.0.0:0.0.0.0, daddr=0.0.0.0:0.0.0.0, sport=0-65535
, dport=0:65535

AWS-fgt1b #
```

Create Internal FortiGate HA Cluster using Template



Deployment Steps in this Demo

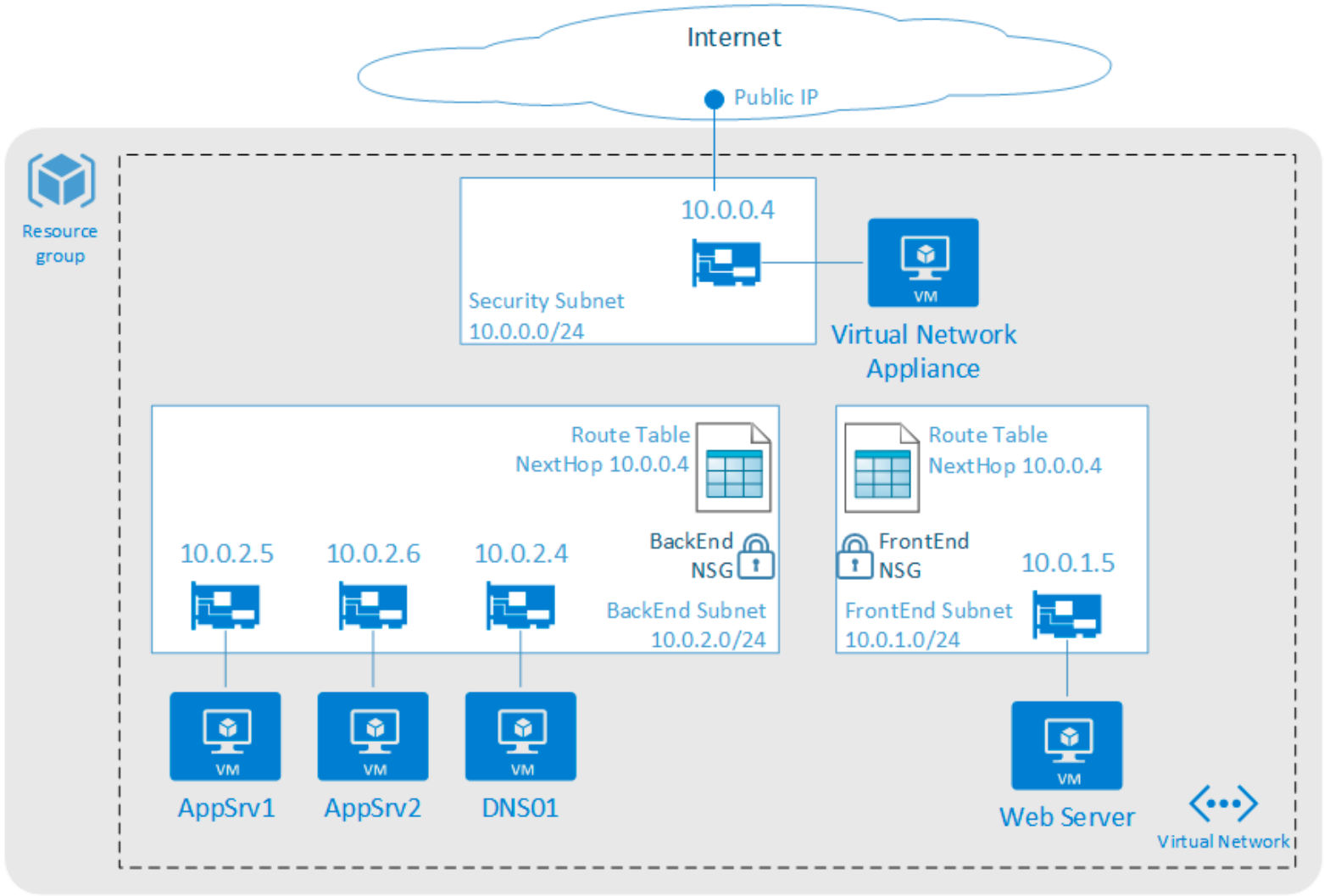
- Create Virtual Network and Subnets
- Deploy External FortiGate HA (A/A) Cluster from Azure Marketplace
- Configure External Load Balancer
- Configure external FortiGates
- Deploy DMZ Server
- Configure Azure load balancer and routes for outbound
- Deploy Internal FortiGate HA (A/A) Cluster from Template
- Configure internal FortiGates from DMZ Host
- Configure Azure internal load balancer and routes

Lessons Learned

- Access internal FortiGates from DMZ server. I've encountered login difficulties when connecting from Internet – I wouldn't get past the login mask.
- Don't use ping to check connectivity to the Internet. It's not supported across Azure load balancers. Use e.g. Telnet instead.

Wrap-Up

Read Microsoft Documentation



https://docs.microsoft.com/en-us/azure/best-practices-network-security

Glossary

- User Defined Routes
  - » Policy routes which can override system and BGP routes on the Azure route service. These are typically required to engage FortiGate in an Azure VNET.
- Network Security Groups
  - » Layer 3/Layer 4 ACLs which can be assigned to NICs or Subnets in Azure. These should not be configured on the same subnet with a UDR and a UDR route target (next hop), so typically not on the same subnet as the FortiGate.
- Availability Set
  - » Provides a model to separate VMs into unique fault domains. This is the reliability model for Azure. FortiGates in HA should be in a common Availability Set

Typical Technical Problems

- Azure Routing
  - » UDRs incorrectly deployed or not applied to appropriate subnet
  - » UDRs and NSGs deployed on the same subnet as the FortiGate
  - » UDRs not actually overriding system or BGP routes because they are not as specific

- FortiGate Routing
  - » FortiGate must have route to any network or subnet in which it doesn't have an interface.
  - » FortiGate route table is distinct and disconnected from Azure routes.


FORTINET